

# RESTRUCTUREDDB: Restructure Database

The RESTRUCTUREDDB function unloads an entire database to a sequential dataset, which can be used as input to the STORE function to load the data into a new database. The target database may be located on a physical device type different from the source database.

```

ADAORD { RESTRUCTUREDDB | REDB }
    [DBASSODEV = { device-type | ADARUN-device } ]
    [DBDATADEV = { device-type | ADARUN-device } ]
    [DBINDEXCOMPRESSION = { YES | NO } ]
    [FILE = file-number ]
    [ASSOPFAC = padding-factor ]
    [ASSODEV = { device-type | DBASSODEV-value } ]
    [DATADEV = { device-type | DBDATADEV-value } ]
    [DATAPFAC = padding-factor ]
    [INDEXCOMPRESSION = { YES | NO } ]
    [ISNSIZE = { 3 | 4 } ]
    [SORTSEQ = { descriptor | ISN | physical-sequence } ]
    [LPB = { prefetch-buffer-size | ADARUN-lu } ]
    [NOUSERABEND]
    [TEST]

```

This chapter covers the following topics:

- Optional Parameters and Subparameters
- Examples

## Optional Parameters and Subparameters

### ASSODEV: Associator Device Type

ASSODEV specifies the device type to be used in the new database for the file's ASSO dataset. This parameter is required only when the device type to be used is different from the default device type. The default device type is specified by the DBASSODEV parameter; if DBASSODEV is not specified, the default is the device type specified by the ADARUN DEVICE parameter. These parameters have no effect on the data written to the DDFILEA/FILEA dataset.

### ASSOPFAC: Associator Padding Factor

ADAORD uses the ASSOPFAC value to calculate the space required to perform the STORE function for the specified file. Valid values are 1-90. The number of AC, NI, and UI blocks is calculated for the device type specified by ASSODEV and the padding factor specified by ASSOPFAC. These parameters have no effect on the data written to DDFILEA. If ASSOPFAC is not specified, the current padding factor for the file is used.

**DATADEV: Data Storage Device Type**

DATADEV specifies the device type to be used for the specified file's new DATA dataset. This parameter is required only when the device type to be used is different from the default device type. The default device type is specified by the DBDATADEV parameter; if DBDATADEV is not specified, the default is the device type specified by the ADARUN DEVICE parameter. These parameters have no effect on the data written to DDFILEA.

**DATAPFAC: Data Storage Padding Factor**

ADAORD uses DATAPFAC to calculate the space required to perform the STORE function for the specified file. Valid values are 1-90 (see the ADALOD LOAD DATAPFAC parameter discussion for more information about setting the padding factor). The number of Data Storage blocks is calculated for the device type specified by DATADEV and the padding factor specified by DATAPFAC. If DATAPFAC is not specified, the current padding factor for the file is used. These parameters have no effect on the data written to DDFILEA.

**DBASSODEV: Default Associator Device Type**

DBASSODEV specifies a default device type for the new ASSO dataset. ADAORD uses the device type specified here to calculate the ASSO space requirements for each restructured file. If DBASSODEV is not specified, the default is the device type specified by the ADARUN DEVICE parameter.

To override the default device type for a file, use the FILE and ASSODEV parameters.

**DBDATADEV: Default Data Storage Device Type**

DBDATADEV specifies a default device type for the new DATA dataset. ADAORD uses the device type specified here to calculate the DATA space requirements for each restructured file. If DBDATADEV is not specified, the default is the device type specified by the ADARUN DEVICE parameter.

To override the default device type for a file, use the FILE and DATADEV parameters.

**DBINDEXCOMPRESSION: Calculate Index Sizes for Database**

DBINDEXCOMPRESSION indicates for all files whether the index space calculation performed and displayed by ADAORD is based on compressed or uncompressed indexes. It applies to all files for which no INDEXCOMPRESSION parameter is specified.

DBINDEXCOMPRESSION can be used to calculate the sizes of compressed or uncompressed indexes for all files of the database, making it unnecessary to calculate the sizes for each file.

**FILE: File Number**

FILE specifies the file to which the following parameters apply. The records for all files not specified by this parameter are unloaded in physical sequence, by file.

If an Adabas checkpoint or security file is specified, do not specify the SORTSEQ parameter.

**INDEXCOMPRESSION: Calculate Index Sizes for File**

INDEXCOMPRESSION indicates for its associated file whether the index space calculation performed and displayed by ADAORD is based on a compressed or uncompressed index.

If INDEXCOMPRESSION is *not* specified

- but the DBINDEXCOMPRESSION parameter is specified for the database as a whole, the default is the database value.
- and DBINDEXCOMPRESSION is also *not* specified, the default is the current compression form of the file.

### **ISNSIZE: 3- or 4-Byte ISN**

ISNSIZE specifies whether ISNs in the file are to be 3 or 4 bytes long. The default is the value currently used for the file; this value is stored in the file control block (FCB).

#### **Note:**

It is not possible to change the ISNSIZE of a physically coupled file using ADAORD.

### **LPB: Prefetch Buffer Size**

LPB specifies the size, in bytes, of the internal prefetch buffer. The maximum value is 32760 bytes. The default depends on the ADARUN LU parameter. ADAORD may reduce a specified LPB value if the LU value is too small.

### **NOUSERABEND: Termination without ABEND**

When an error is encountered while the function is running, the utility prints an error message and terminates with user ABEND 34 (with a dump) or user ABEND 35 (without a dump).

If NOUSERABEND is specified, the utility will *not* ABEND after printing the error message. Instead, the message "utility TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

### **SORTSEQ: File Processing Sequence**

SORTSEQ determines the sequence in which the file is to be processed. If this parameter is omitted, the records are processed in physical sequence.

If a descriptor is specified, the file is processed in the logical sequence of the descriptor values. *Do not* use a null-suppressed descriptor field, a hyperdescriptor, a phonetic descriptor, a multiple-value field, or a descriptor contained in a periodic group.

#### **Note:**

Even when the descriptor field is not null suppressed, the record is *not* represented in the inverted list if the descriptor field or a field following it has never been initialized (held a value). Therefore, the record will be dropped when the utility is executed.

If ISN is specified, the file is processed in ascending ISN sequence. For the Adabas checkpoint or security file, only SORTSEQ=ISN is allowed.

### **TEST: Test Syntax**

This parameter tests the operation syntax without actually performing the operation. Only the syntax of the specified parameters can be tested; not the validity of values and variables.

## Examples

### Example 1:

```
ADAORD RESTRUCTUREDDB
```

The RESTRUCTUREDDB function is to be executed. All files are to be unloaded in physical sequence.

### Example 2:

```
ADAORD RESTRUCTUREDDB FILE=146,SORTSEQ=MZ  
ADAORD FILE=151,SORTSEQ=TF
```

The RESTRUCTUREDDB function is to be executed. File 146 is to be unloaded in the sequence of descriptor MZ. File 151 is to be unloaded in the sequence of descriptor TF. All other files are to be unloaded in physical sequence.